Blue Gene System Software: Let's collaborate!

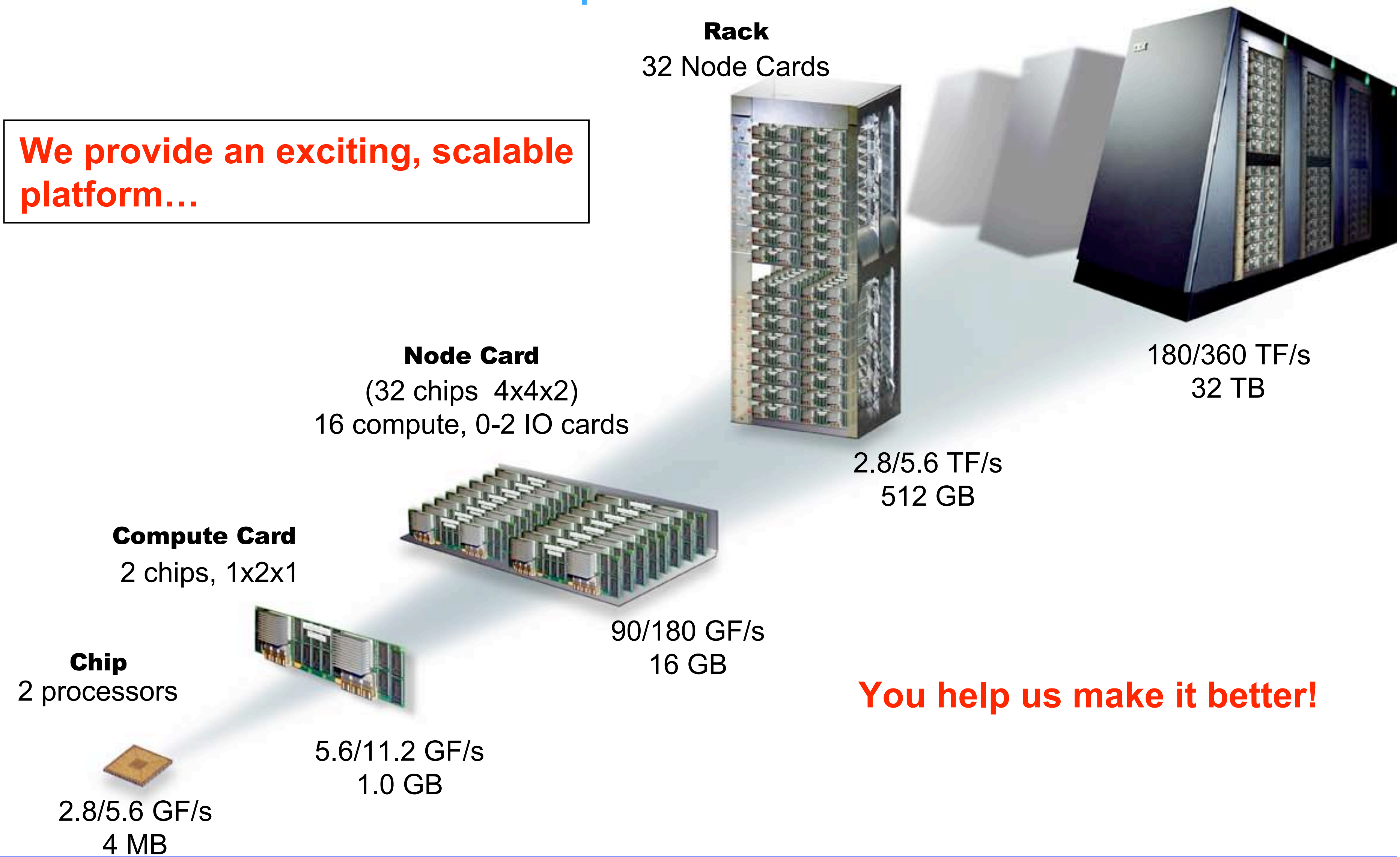
Manish Gupta

IBM Thomas J. Watson Research Center

© 2004 IBM Corporation

Blue Gene Partnership

We provide an exciting, scalable platform…



You help us make it better!

Blue Gene Partnership Goals

- Push system scalability to unprecedented levels
- Support high productivity – make system easier to use and manage
- Make system useful for a broader class of applications

© 2004 IBM Corporation

Blue Gene Partnership Goals

- Push system scalability to unprecedented levels
- Support high productivity – make system easier to use and manage
- Make system useful for a broader class of applications

- Impact on Blue Gene/L product
  - Constraints – supporting changes to base software
  - Opportunities – many areas to augment IBM offering

© 2004 IBM Corporation

Blue Gene Partnership Goals

- Push system scalability to unprecedented levels
- Support high productivity – make system easier to use and manage
- Make system useful for a broader class of applications

- Impact on Blue Gene/L product
  - Constraints – supporting changes to base software
  - Opportunities – many areas to augment IBM offering

- Impact on Blue Gene/P design

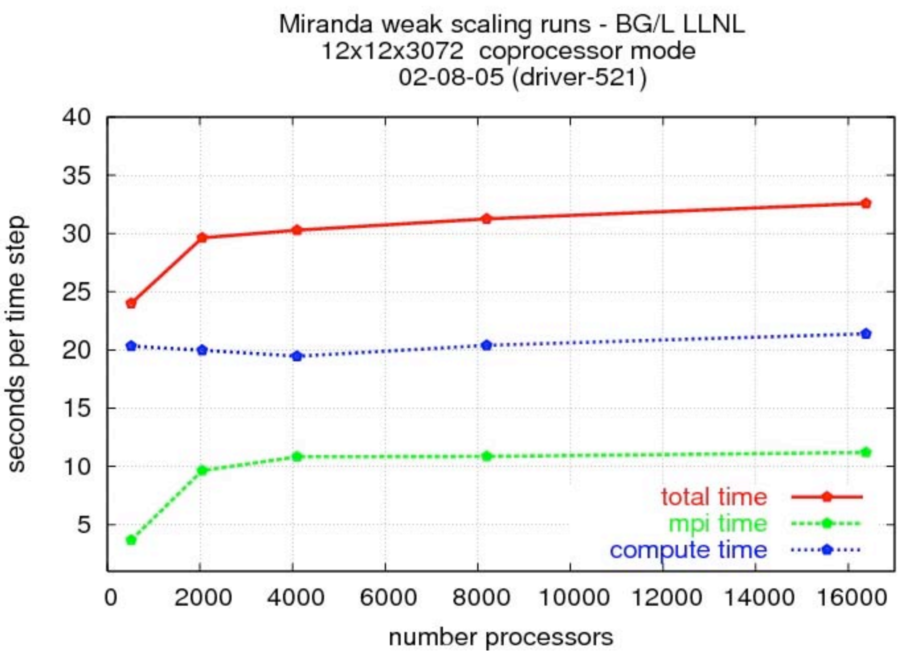
© 2004 IBM Corporation

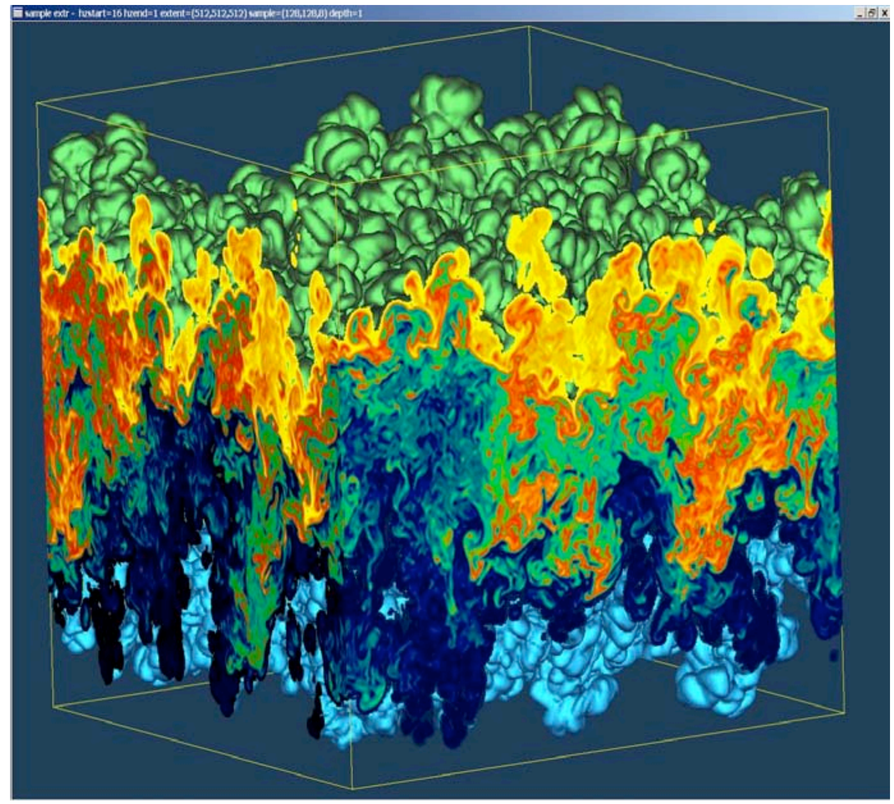
Status Summary

- 16 racks (16,384 nodes, 32768 processors) at Rochester and LLNL
  - Another 16 racks on LLNL floor
- 70.72 TF/s sustained Linpack
  - #1 on TOP500 list
- Various applications and benchmarks executed – IBM and LLNL
  - Highest ever delivered performance on many applications

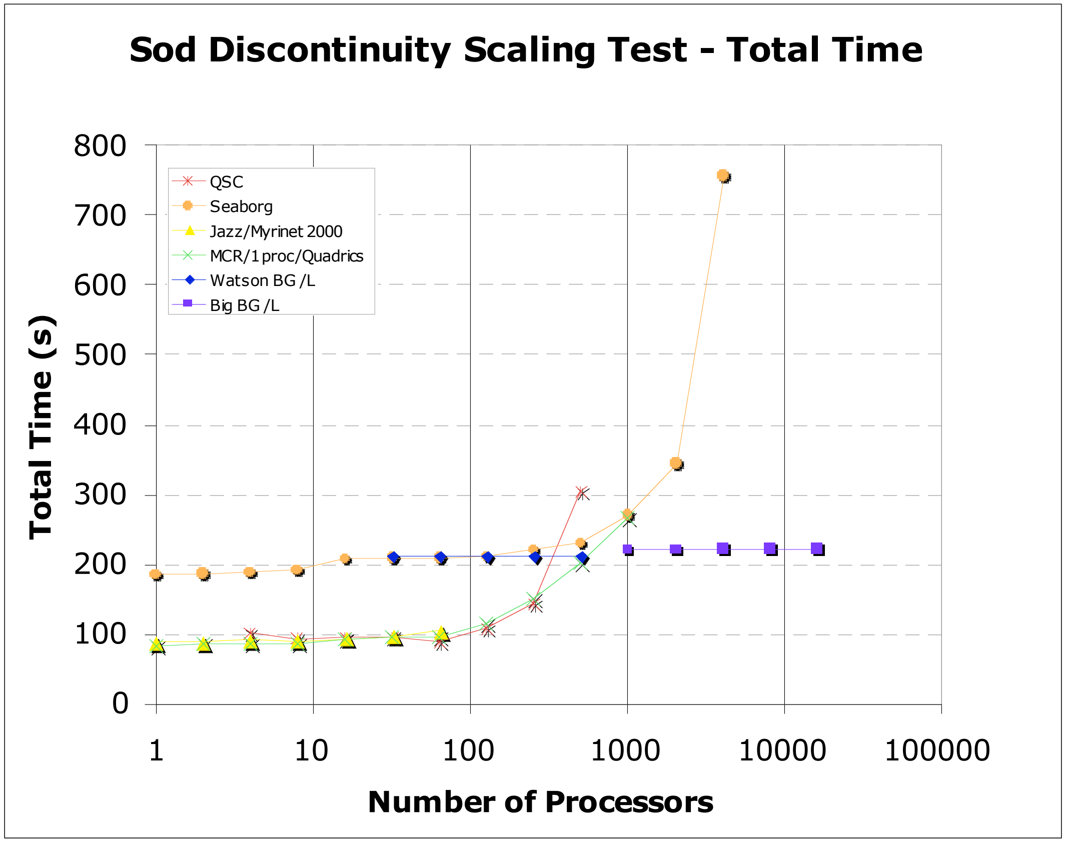
© 2004 IBM Corporation

Miranda Weak Scaling on BG/L





FLASH: Astrophysics Code from Argonne National Lab
SCALING TO 16x1024 nodes on Blue Gene/L



Big BGL: 16 Racks, coprocessor, 440

Jazz : 350node, 2.4GHz Xeon, ANL

MCR : 1152node,2.4GHz Xeon,LLNL

Seaborg: IBM SP, 1.5GF/node NERSC

QSC: 256nodex4way HP Alpha, LLNL

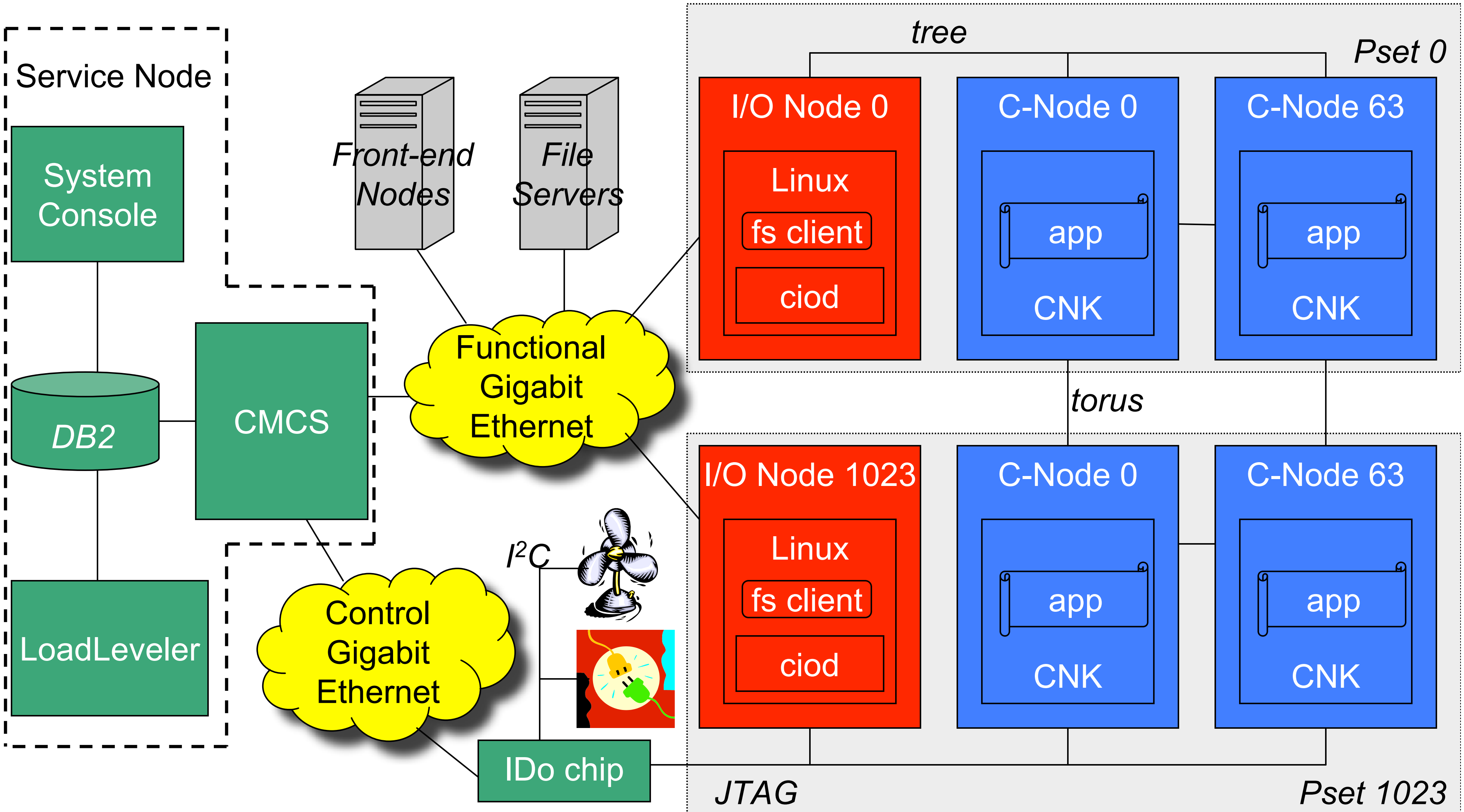
© 2004 IBM Corporation

Much work in progress…

- Parallel file system (GPFS) under installation and test
- Job scheduling solution (LoadLeveler) coming soon
- System management enhancements
- MPI enhancements
- Math libraries (full ESSL, MASS, MASSV) being developed
- Performance tools being developed
- Compiler enhancements

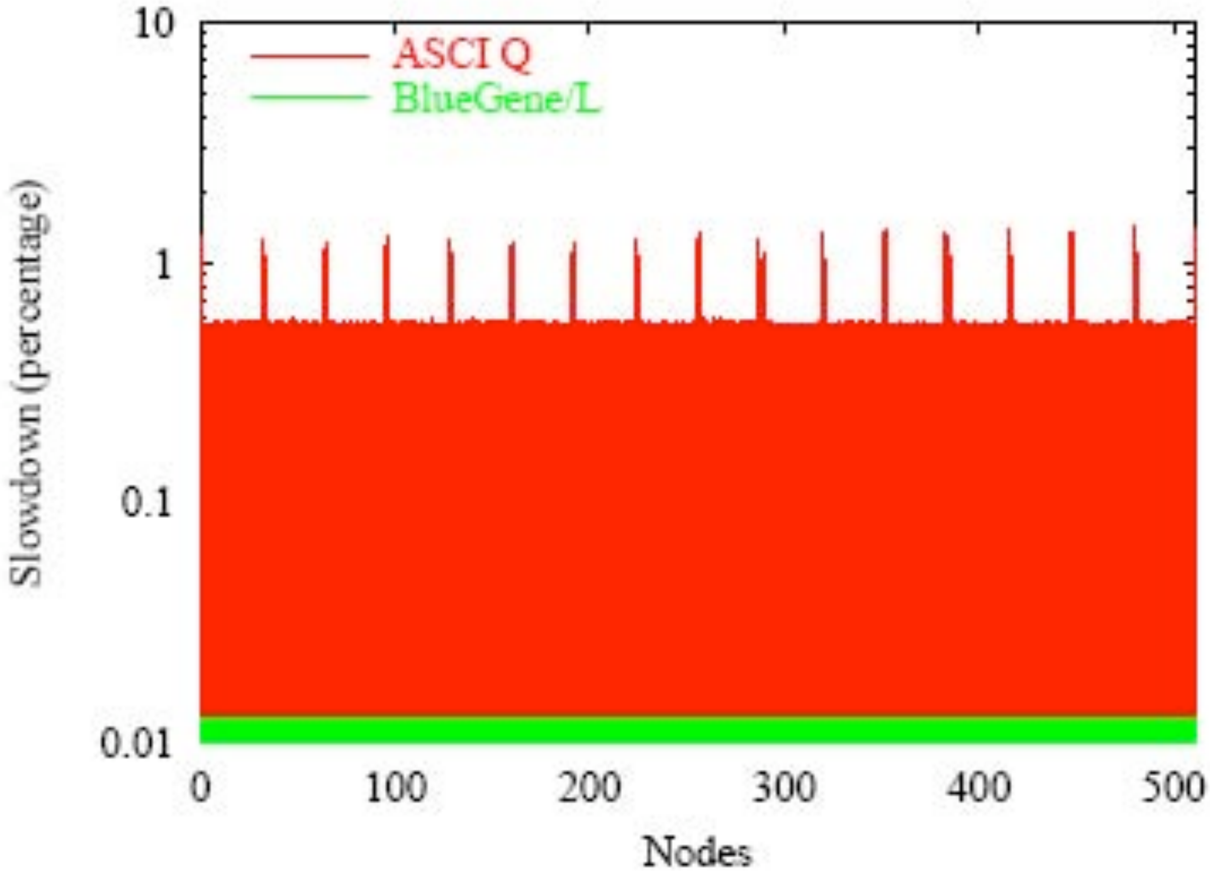
© 2004 IBM Corporation

BlueGene/L System Architecture



© 2004 IBM Corporation

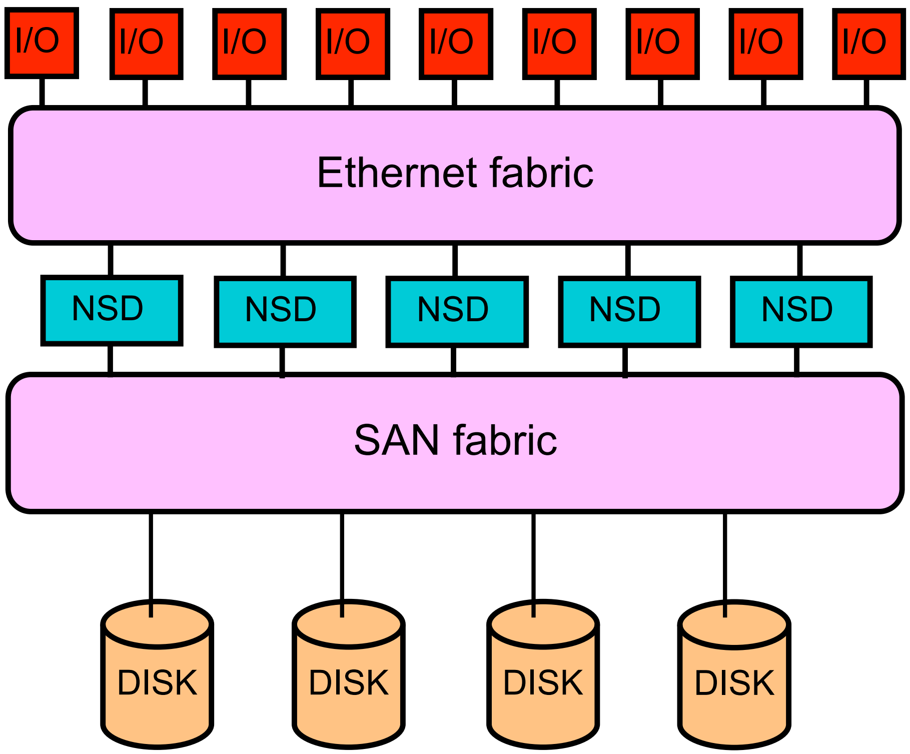
Noise measurements (from Adolphy Hoisie)



Ref: Blue Gene: A Performance and Scalability Report at the 512-Processor Milestone, PAL/LANL, LA-UR- 04-1114, March 2004.

© 2004 IBM Corporation

Parallel File System for BlueGene/L (GPFS)



- GPFS solution for BlueGene/L is 3-tiered
  - First tier consists of the I/O nodes, which are GPFS clients – currently run NFS clients
  - Second tier is a cluster of NSD (Network Shared Disk) servers
  - Third tier is a set of storage devices, typically fiber channel or iSCSI
- First-to-second tier interconnect has to be Ethernet
- Second-to-third tier can be fiber channel loop, fiber channel switch, or Ethernet (for iSCSI)
- Choice of NSD servers, SAN fabric and storage devices depends on specific requirements

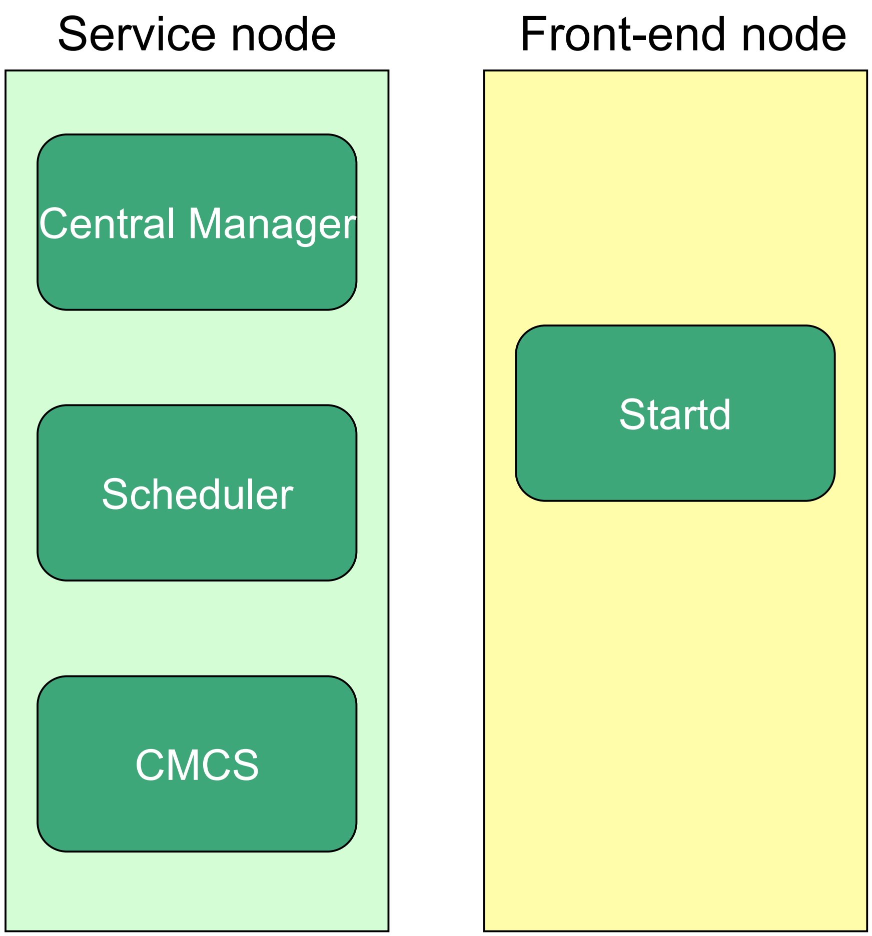
 © 2004 IBM Corporation

Job Scheduling in BlueGene/L

- LoadLeveler solution
  - BG/L specific job scheduler plugged into LoadLeveler as external scheduler
  - Working on a integrated, internal scheduler, solution
- Job scheduling strategies can significantly impact the utilization of large computer systems
  - Machines with toroidal topology (as opposed to all-to-all switch) are particularly sensitive to job scheduling – this was demonstrated at LLNL with gang scheduling on Cray T3D
  - BG/L scheduling strategies leveraging BG/L unique topology features can significantly enhance system utilization – from 45% to almost 90% (depends on workload)

© 2004 IBM Corporation

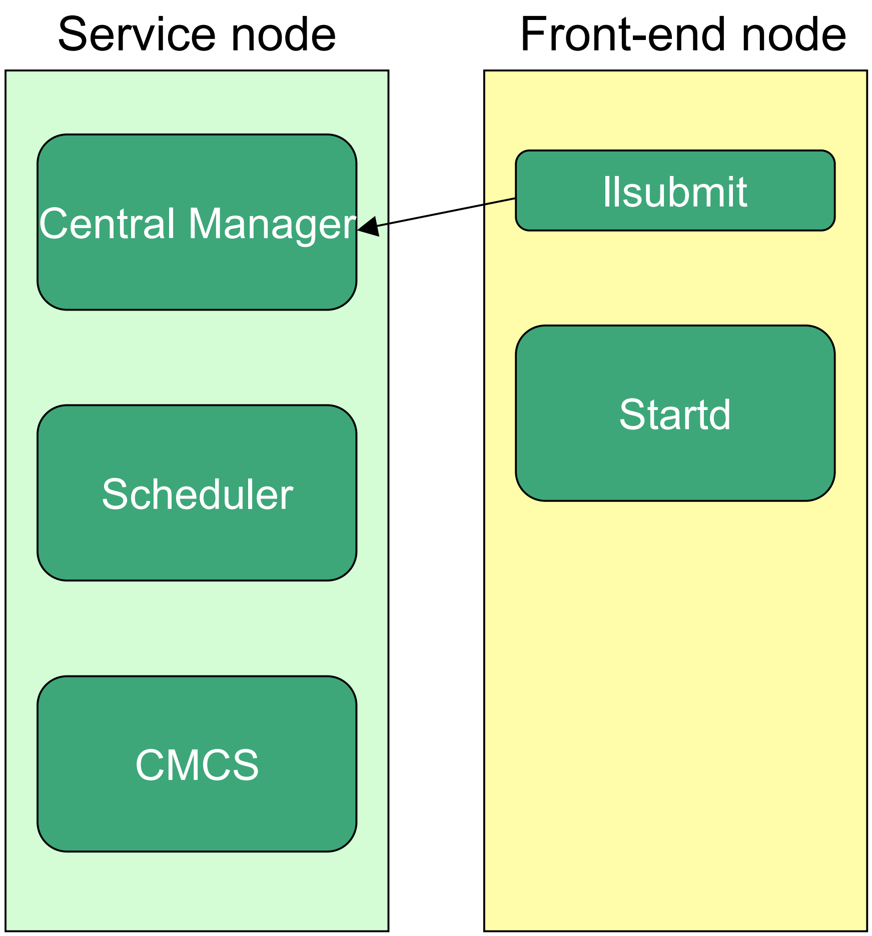
LoadLeveler for BlueGene/L



- The BlueGene/L implementation of LoadLeveler is contained entirely in the service and front-end nodes
- The service node runs the *Central Manager* daemon and external scheduler
- Front-end nodes run the *Startd* daemon

© 2004 IBM Corporation

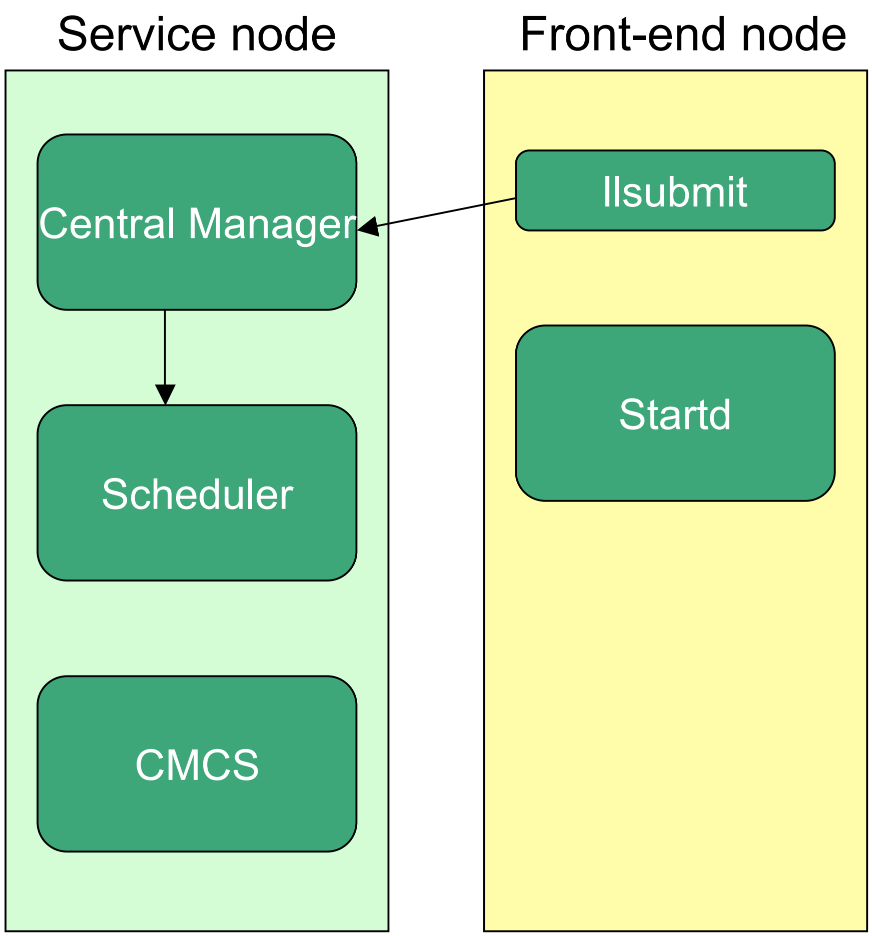
LoadLeveler for BlueGene/L



- The user submits a job from the front-end node
- The llsubmit command contacts the Central Manager to enqueue the job for executions

© 2004 IBM Corporation

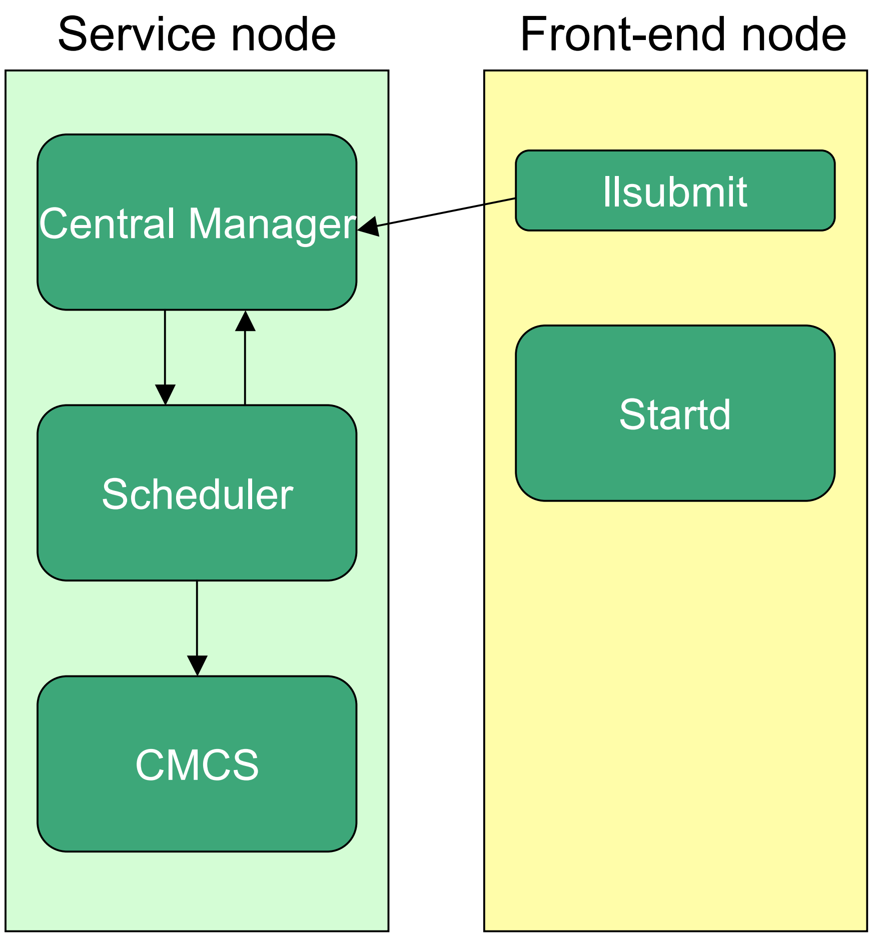
LoadLeveler for BlueGene/L



- The user submits a job from the front-end node
- The llsubmit command contacts the Central Manager to enqueue the job for executions
- The scheduler retrieves the queue of jobs to execute and makes policies decisions

 © 2004 IBM Corporation

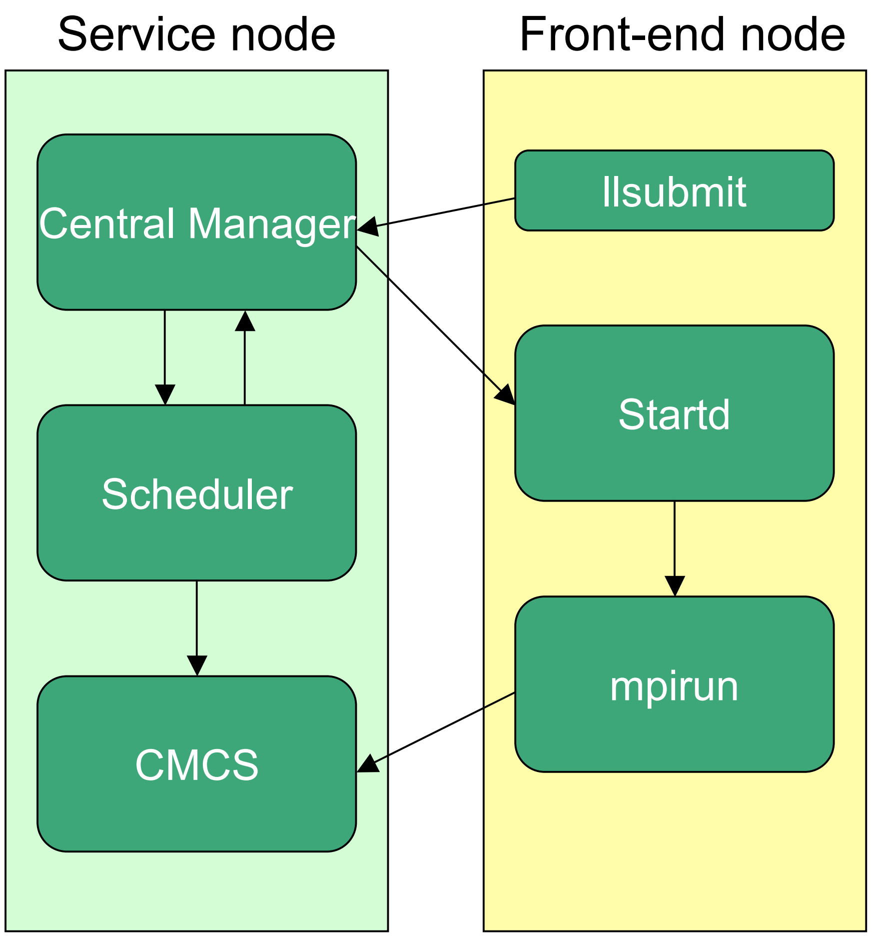
LoadLeveler for BlueGene/L



- The user submits a job from the front-end node
- The llsubmit command contacts the Central Manager to enqueue the job for executions
- The scheduler retrieves the queue of jobs to execute and makes policies decisions
- The scheduler uses control system services to create a machine partition and instructs the Central Manager to start the job

 © 2004 IBM Corporation

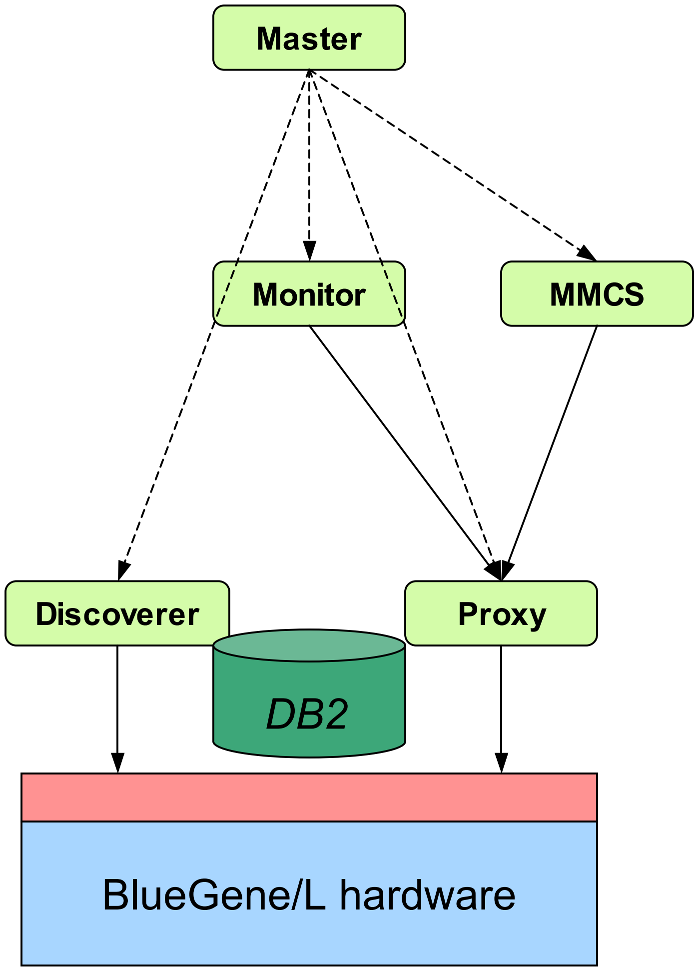
LoadLeveler for BlueGene/L



- The Central Manager contacts the Startd daemon on the front-end node to launch mpirun
- The mpirun process uses control system services to launch the actual application processes in the partition created by the scheduler
- The mpirun process stays in the front-end node as a proxy of the user application
- Debuggers (e.g., TotalView) work by attaching to the mpirun process

© 2004 IBM Corporation

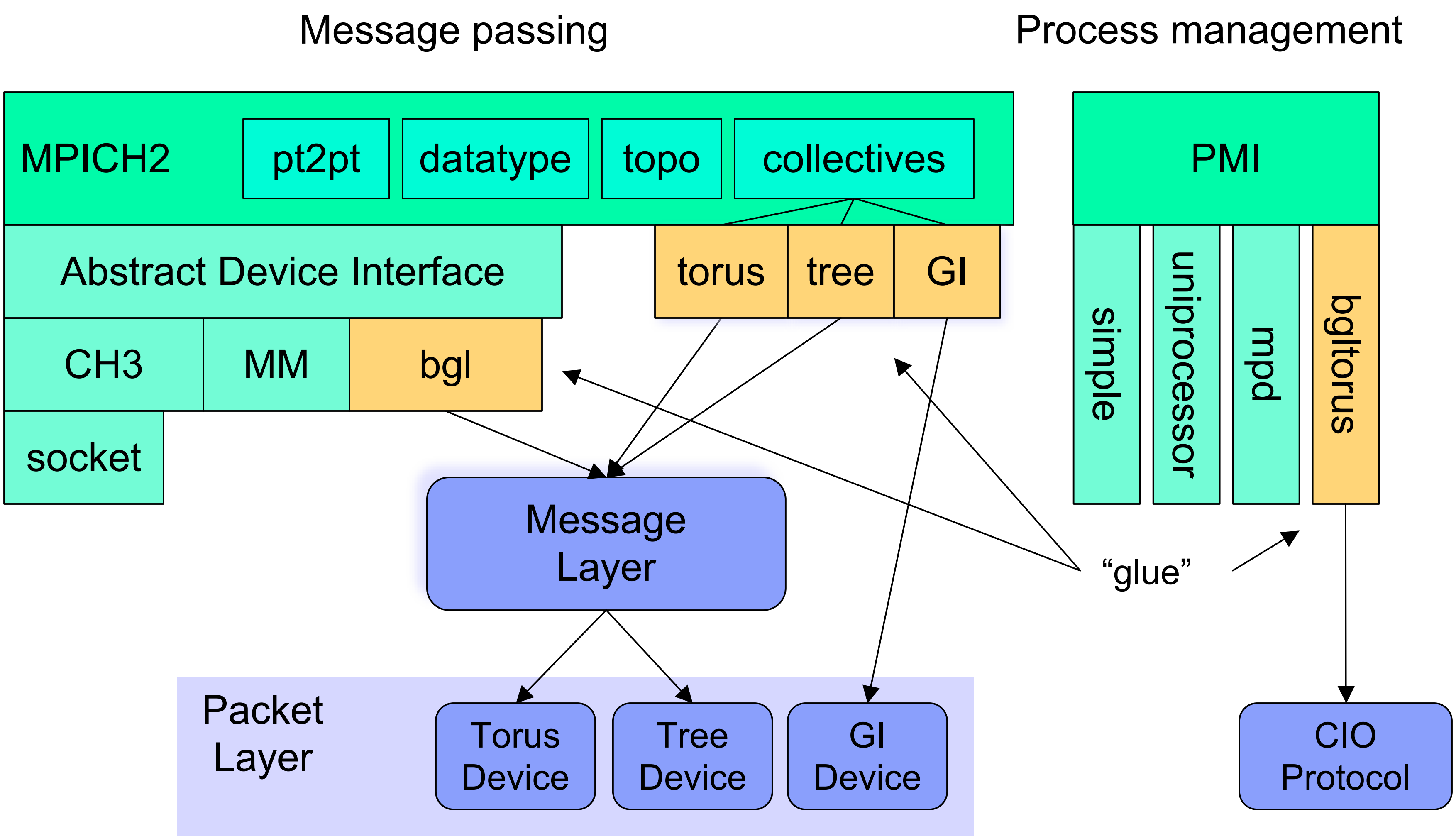
Control System – Components



- Master creates, monitors, and restarts the other processes
- Discoverer finds and initializes new hardware
- Proxy virtualizes the IDo hardware, providing reliable and atomic connection
- Monitor monitors environmentals, such as temperature and voltages
- MMCS configures and IPLs partitions of the machine, bringing those partitions to a user-architected state

© 2004 IBM Corporation

MPI – based on MPICH2 from ANL



© 2004 IBM Corporation

MPI enhancements

- Higher levels of scalability
  - Continued enhancements of collectives
  - Adaptive buffer management with flow control
  - Support for interrupts
  - Adaptive protocol selection with compiler analysis
- MPI-IO support
  - BG/L specific optimizations
  - Optimize GPFS based on higher level view

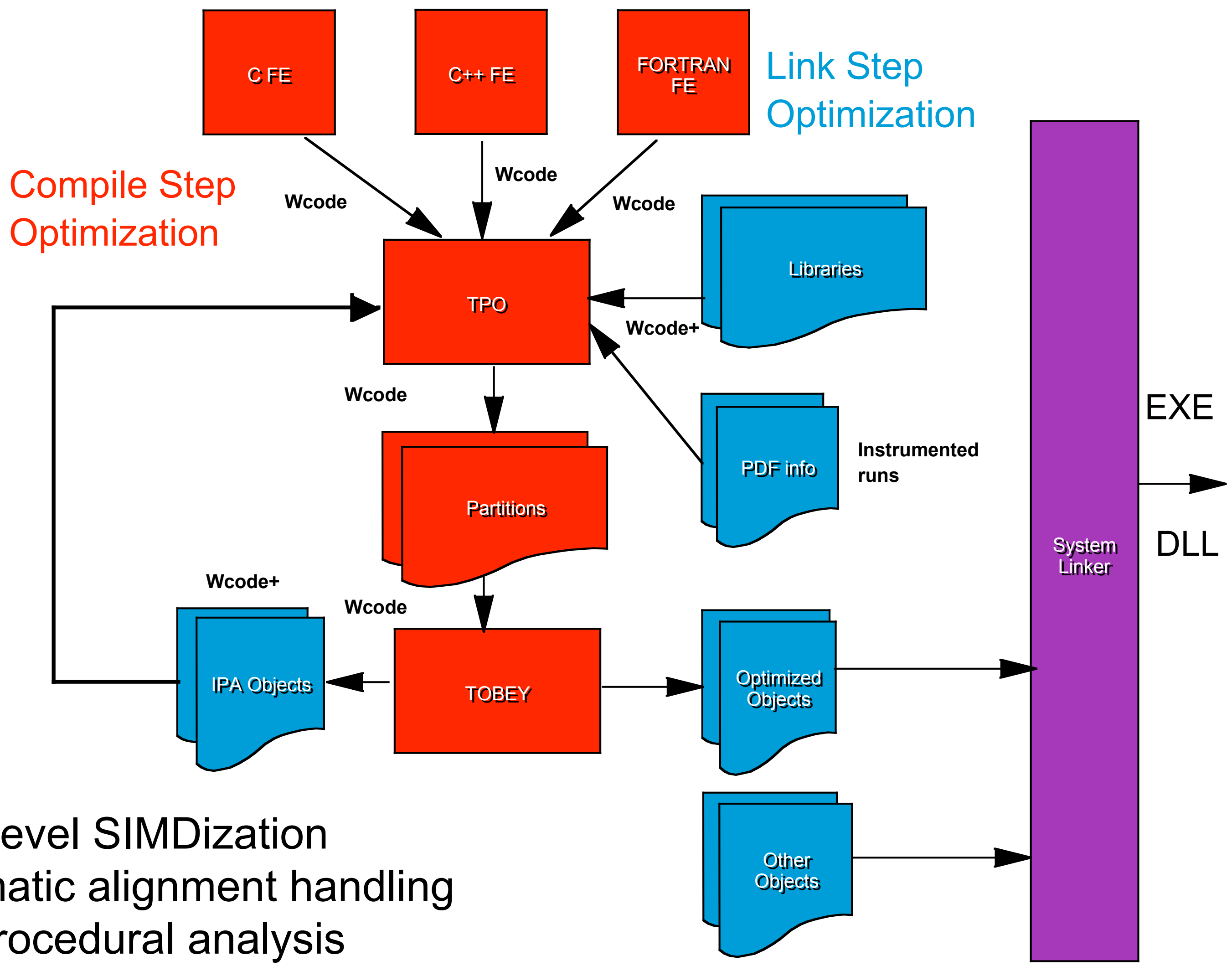
© 2004 IBM Corporation

Strategy to Exploit SIMD FPU

- Automatic code generation by compiler (-qarch=440d)
  - Single FPU fallback: -qarch=440
- User can help the compiler via pragmas and intrinsics
  - Pragma for data alignment: *__alignx(16, var)*
  - Pragma for parallelism
    - Disjoint: *#pragma disjoint (*a, *b)*
    - Independent: *#pragma ibm independent* loop
  - Intrinsics
    - Intrinsic function defined for each parallel floating point operation
      - E.g.: *D = __fpmadd(B, C, A) => fpmadd rD, rA, rC, rB*
    - Control over instruction selection, compiler retains responsibility for register allocation and scheduling
- Using library routines where available
  - Dense matrix BLAS – e.g., DGEMM, DGEMV, DAXPY
  - FFT
  - MASS, MASSV

© 2004 IBM Corporation

IBM Compiler Architecture



Loop level SIMDization
Automatic alignment handling
Interprocedural analysis

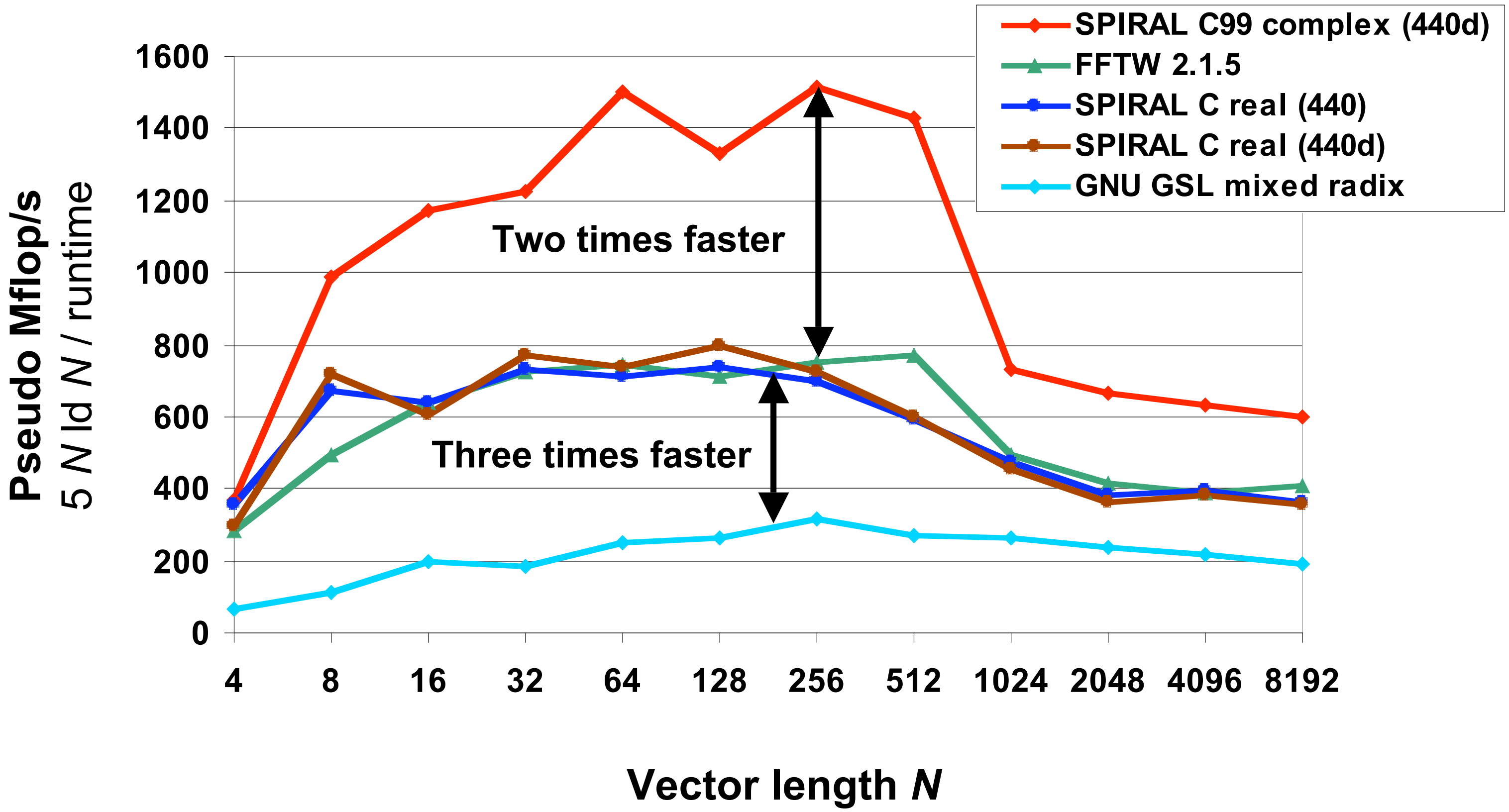
© 2004 IBM Corporation

Math Libraries: ESSL

- Started with small subset (of ~500 routines)
  - Mainly dense matrix kernels – DGEMM, DGEMV, DDOT, DAXPY etc.
- Using ESSL source code to drive compiler testing and exploration of complete ESSL support
  - Status: Nearly complete functionality available using –O3 –qarch=440
  - Currently investigating SIMD FPU issues, performance enhancements
  - Expected general availability – Nov 2005
- FFT
  - Technical University of Vienna developing FFT library optimized for BlueGene/L – effective use of the SIMD FPU

© 2004 IBM Corporation

FFT Measured Performance



DFT 2^n, complex, double precision

VisualAge XL C 7.0 for BlueGene/L options: -O3 qnostrict – qarch=440/440d

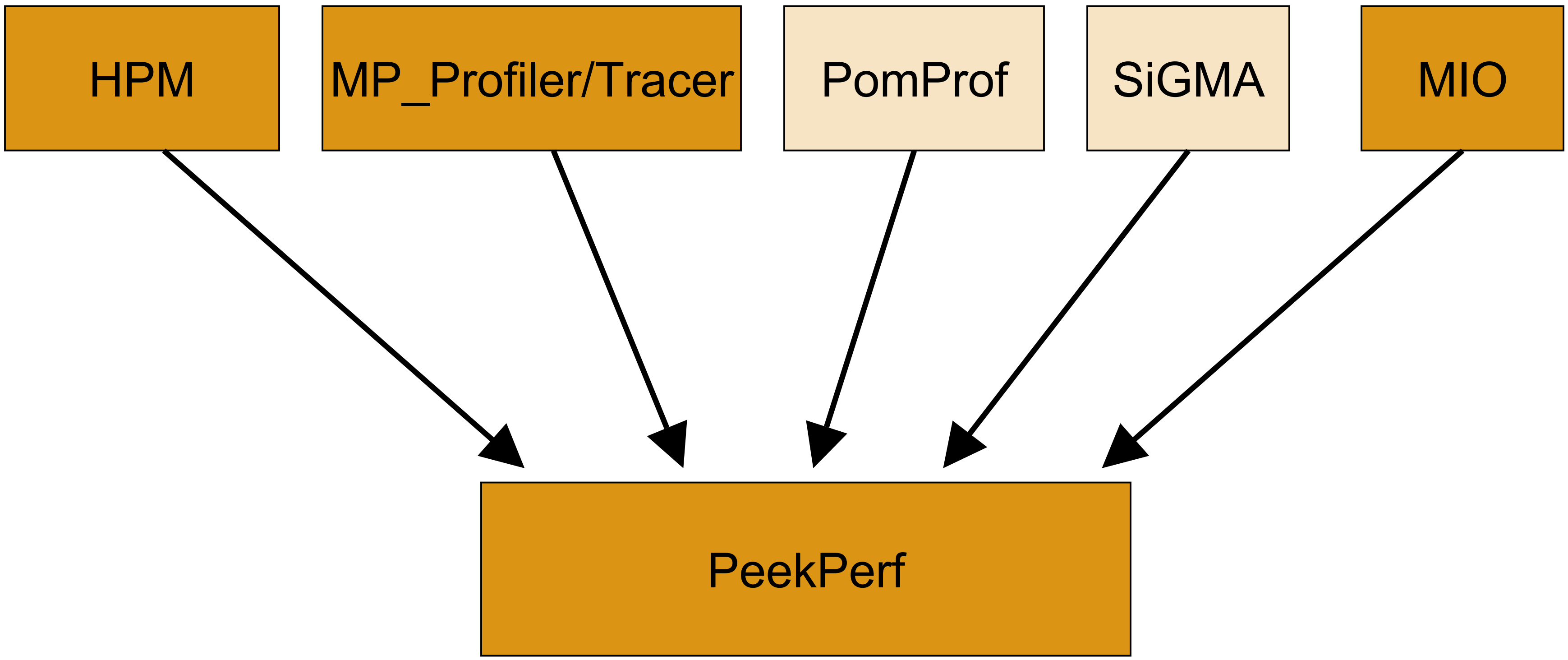
BlueGene/L DD2 prototype at IBM T.J. Watson Research Center
Single BlueGene/L CPU at 700 MHz (one Double FPU)

Math Libraries: MASS and MASSV

- Math intrinsic routines – e.g., square root, exponential, sine, cosine (~50 routines)
  - Traditionally supported on pSeries platforms with hand-tuned assembler routines
  - Up to factor of 5-20x performance boost over naïve versions
- BG/L: Novel approach using special compilation of versions written in C
  - Being deployed by Toronto compiler team on Apple platform
  - Complete set of routines available using this approach
  - Reciprocal, square root, reciprocal square root, exponential, logarithm, cube root optimized for BG/L – prioritized based on early applications
  - Expected availability of MASS, MASSV – June 2005

© 2004 IBM Corporation

Performance Tools – based on IBM HPCT



Additional tools - Code profiler (gprof, Xprofiler), Mapping tool for 3D torus topology
New challenge – scalability of tools

© 2005 IBM Corporation

Advanced Programming Models

- Global Arrays
  - Prototype implementation of ARMCI (active message library) on BG/L
    - ARMCI used as a driver for active message libraries
      - » Motivated a rewrite of message layer
  - Performance problems in handling Torus interrupts
    - >10000 cycles currently
  - Prototyping new message layer to provide interoperability between MPI and ARMCI

- UPC
  - Pursued as part of PERCS project
    - Extensive work on front end and compiler at Toronto
  - Port of UPC runtime to Blue Gene feasible

- MATLAB-like environment for linear algebra
  - Collaboration with UIUC

Collaborations: Improving Programmer Productivity

- High performance libraries and packages
  - Computation – ScaLAPACK, sparse matrix BLAS, PDE solvers, PETSc, …
  - I/O – parallel netCDF, parallel HDF5 libraries
    - MPI-IO optimizations

- Performance tools
  - Identification of performance bottlenecks
  - Techniques for scalability

- Programming models
  - MPI enhancements – topology awareness, fault tolerance
  - Global address support – Global Arrays, UPC, Co-Array Fortran

© 2004 IBM Corporation

Conclusions

- Blue Gene/L represents a new level of performance scalability and density for scientific computing
- Blue Gene/L system software stack with Linux-like personality for applications
  - Custom solution (CNK) on compute nodes for highest performance
  - Linux solution on I/O nodes for flexibility and functionality
  - MPI is the default programming model, others are being investigated
- Encouraging performance results – excellent scaling to 16K nodes
- Great opportunities for collaboration
  - Complement IBM efforts on BG/L
  - Impact BG/P design

© 2004 IBM Corporation